Towards Recursive Self-Evolving Agentic Literature Retrieval

Yuwen Du[1,2*†], Tian Jin[1,2†], Jing Kang[1,2], Xianghe Pang[1,2], Jingyi Chai[1,2], Tingjia Miao[1,2], Fenyi Liu[1,2], WenHao Wang[3], Sikai Yao[2], Yuzhi Zhang[2], Siheng Chen[1,2*]

[1]Shanghai Jiao Tong University, Shanghai, China.
[2]SciLand, Shanghai, China.
[3]Zhejiang University, Hangzhou, China.

*Corresponding author(s). E-mail(s): sjtu0267098@sjtu.edu.cn; sihengc@sjtu.edu.cn;
†These authors contributed equally to this work.

Abstract

Scientific literature retrieval must understand complex search intents while preserving source authenticity. Traditional keyword and embedding-based systems return authentic sources but miss nuanced intents, whereas large language models capture richer intents but may fabricate citations. We introduce PaSaMaster, a Recursive Self-Evolving agentic literature retrieval system that iteratively analyzes intent, retrieves verified papers and ranks them with evidence-grounded relevance scores. PaSaMaster combines self-evolving retrieval that refines search intent from ranked evidence over time, hallucination-free ranking over verified papers rather than generated citations, and cost-efficient planning–retrieval separation that reserves frontier LLMs for intent understanding while delegating retrieval and scoring to lightweight models and customized corpora. Across 38 disciplines in PaSaMaster-Bench, PaSaMaster achieves a 16.5× higher F1-score than Google Scholar and a 37.8% higher F1-score than GPT-5.2 at about 1% of the cost, while reducing source hallucination from 32.66% in generative LLMs to zero: https://github.com/sjtu-sai-agents/PaSaMaster

Keywords: Scientific Literature Discovery, Agentic AI, Large Language Models

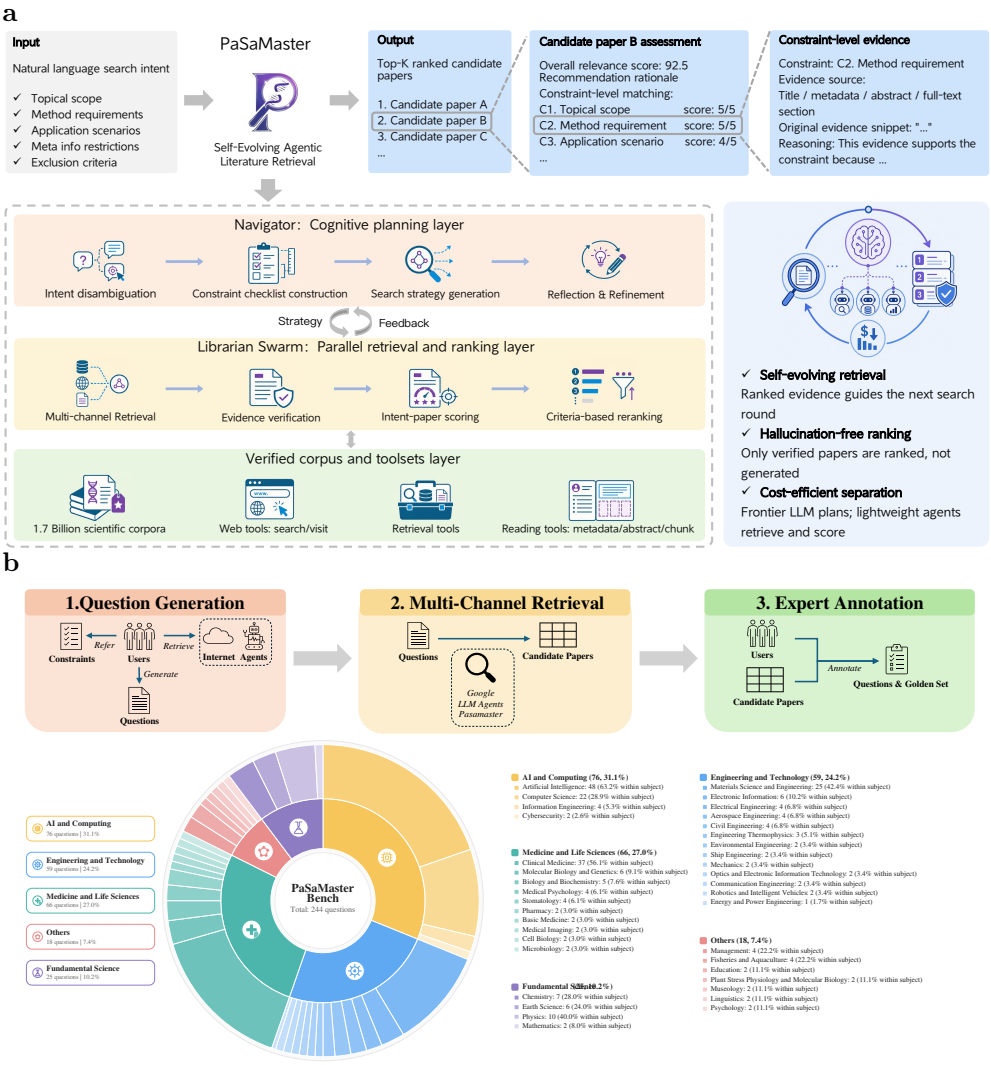

Fig. 1 Overview of PaSaMaster and PaSaMaster-Bench. a, PaSaMaster converts a complex natural-language search intent into a verified, evidence-ranked paper list through three layers: a Navigator that refines intent and search strategy, a Librarian Swarm that retrieves, verifies and scores candidate papers, and verified corpora and tools that ground all outputs in real sources. **b,** PaSaMaster-Bench evaluates complex literature retrieval through expert-written search intents, multi-channel candidate retrieval and checklist-based expert annotation. The benchmark contains 244 tasks across 38 disciplines, enabling evaluation of intent understanding, source reliability and ranking quality.

Table 1 The Five Paradigms of Scientific Literature Discovery. Existing paradigms each improve one aspect of literature retrieval, but fail to jointly achieve adaptive intent understanding, hallucination-free evidence grounding, and cost-efficient scaling. PaSaMaster resolves these limitations through agentic Recursive Self-Evolving, evidence-grounded retrieval.

Paradigm Level	Representative Systems	Intent Adaptivity	Source Reliability	Cost Efficiency
Level 0: Lexical Retrieval	Google Scholar [1], PubMed [2]	Keyword-based; severe intent compression	Verified indexed papers	Efficient but semantically shallow
Level 1: Semantic Retrieval	OpenScholar [3], Bohrium Navigator [4]	Passive embedding matching; limited intent compression	Verified indexed papers	Efficient but semantically shallow
Level 2: Generative LLMs	GPT-5.2 [5], Gemini 3.1 Pro [6], DeepSeek [7]	Strong natural-language understanding	Prone to hallucinated papers	Expensive to deploy at scale
Level 3: Fixed-Pipeline Agentic Retrieval	Google Scholar Labs [8], PaSa [9]	User intent fixed at the outset; no cognition update during retrieval	Verified indexed papers	Cost-controlled, but constrained by fixed intent interpretation
Level 4: Recursive Self-Evolving Agentic Retrieval	**PaSaMaster (Ours)**	**Iteratively refines intent using ranked evidence**	**Verified indexed papers**	**Cost-efficient planning–retrieval separation**

Scientific literature retrieval is the axiomatic starting point of all scientific inquiry [10]. Before formulating hypotheses, designing experiments, or building new theories, researchers must fundamentally navigate the vast and ever-expanding corpus of existing knowledge [11]. However, the volume of scientific publications has grown exponentially over recent decades, decisively overwhelming the fixed cognitive bandwidth of individual researchers [12–14]. This severe information overload has driven an inevitable reliance on artificial intelligence to automate and accelerate knowledge discovery [15]. More importantly, modern literature search is rarely a simple keyword lookup [16, 17]. Researchers often express complex academic intents involving technical constraints, application contexts, and implicit background knowledge [10, 18, 19]. As large language models reshape scientific research workflows, literature retrieval therefore faces a new central challenge: *how to deeply understand complex search intents while ensuring that every returned source is real and verifiable* [19, 20].

Existing literature retrieval systems still struggle to jointly achieve complex intent understanding and source authenticity. Some methods preserve source authenticity at the cost of shallow intent understanding [1, 3, 4], whereas others improve semantic comprehension while sacrificing factual reliability [5–7, 21–23]. This creates a persistent tradeoff between reliable but limited retrieval and more intelligent but less trustworthy literature discovery.

The tradeoff between intent understanding and source reliability is clearer when viewed through the evolution of literature retrieval paradigms (Table 1). *Level 0 (Lexical Retrieval)* [1, 2] guarantees source authenticity through indexed databases, but reduces complex research intents to rigid keywords, causing severe intent compression. *Level 1 (Semantic Retrieval)* [3, 4] improves over exact keyword matching by using embedding-based similarity and retrieval-augmented matching [24, 25], but still treats retrieval as passive query–document matching and lacks the ability to actively clarify, decompose, or refine complex intents. *Level 2 (Generative LLMs)* [5–7, 21–23] offers stronger intent comprehension, yet its probabilistic generation introduces fabricated papers [26, 27], undermining the factual trust required for scientific inquiry. *Level 3 (Fixed-Pipeline Agentic Retrieval)* [8, 9] mitigates source hallucination by grounding

LLM agents in verifiable retrieval tools [28–33]. However, these systems typically follow a predefined retrieve–read–answer pipeline in which the interpretation of the user question is largely fixed at the beginning. For complex research intents, this initial interpretation can be incomplete or biased toward only part of the request, causing later retrieval steps to miss relevant subtopics or return papers that satisfy only some constraints. This limitation motivates retrieval systems that can update their understanding of the intent as evidence accumulates.

The unresolved need for adaptive, verifiable and efficient retrieval motivates *Level 4 (Recursive Self-Evolving Agentic Retrieval)*, represented by **PaSaMaster**. PaSaMaster is a Recursive Self-Evolving agentic literature retrieval system that iteratively analyzes intent, retrieves verified papers and ranks them with evidence-grounded relevance scores. Rather than treating literature search as one-shot query–document matching problem, PaSaMaster formulates scientific literature discovery as a Recursive Self-Evolving intent–paper relevance ranking process. This design enables the system to align with complex research intents while ensuring that every returned source is real, verifiable, and grounded in customized corpora.

PaSaMaster is built on three key designs. First, Recursive Self-Evolving retrieval: it transforms literature retrieval from one-shot query–document matching into an adaptive search process that evolves over time [34, 35], where retrieved and ranked evidence is used to identify coverage gaps, refine the research intent, and guide subsequent retrieval rounds. Second, hallucination-free ranking: it treats literature discovery as intent–paper relevance ranking rather than generation, and trains a lightweight Ranker on multidisciplinary query–paper evidence so that Librarian agents can make expert-style relevance judgments across disciplines while ranking only verified papers grounded in original evidence. Third, cost-efficient planning–retrieval separation: it uses frontier LLMs only for intent understanding and refinement, while delegating large-scale retrieval and relevance scoring to customized scientific corpora and lightweight models. Together, these designs enable PaSaMaster to align with complex research intents while maintaining source verifiability and cost-efficient scalability.

To evaluate retrieval capability on complex natural-language literature search problems, we introduce **PaSaMaster-Bench**, the first multidisciplinary literature retrieval benchmark designed for complex search intents. Unlike conventional retrieval benchmarks [9, 19, 36–38] built around short keyword queries, PaSaMaster-Bench focuses on highly specific, multi-constrained natural language search intents that require systems to search, verify, and rank all papers satisfying explicit criteria. The benchmark contains *244 expert-curated tasks* spanning *38 scientific disciplines*, with queries, constraints, target paper lists, and evaluation checklists annotated and verified by human domain experts.

The PaSaMaster-Bench evaluation reveals the severe inaccuracy and incompleteness of traditional keyword retrieval, with PaSaMaster improving F1-score by 16.5×. The evaluation also exposes the unreliability of generative LLMs, which exhibit hallucination rates up to 32.66%. Remarkably, PaSaMaster outperforms GPT-5.2 by 37.8% while using only 1% of its computational cost, and maintains zero source hallucination. These results demonstrate that Recursive Self-Evolving, evidence-grounded relevance

ranking can improve complex intent understanding, eliminate source hallucination and enable low-cost scientific literature discovery at scale.

1 Results

1.1 PaSaMaster system overview

Figure 1a summarizes how PaSaMaster converts a natural-language search intent into an auditable ranked paper list. The system first makes the implicit structure of the request explicit, turning topical scope, methodological requirements, application context, metadata restrictions and exclusion rules into a retrieval strategy and verification checklist. It then returns top-K candidate papers with paper-level relevance scores, recommendation rationales and constraint-level judgments. Each judgment is linked to the evidence used to make it, including metadata, abstracts, full-text snippets and the reasoning connecting the evidence to the stated constraint.

PaSaMaster is organized into three layers. The Navigator performs intent disambiguation, checklist construction, search planning and round-by-round reflection. The Librarian Swarm executes multi-channel retrieval, evidence verification, intent–paper scoring and criteria-based reranking. The verified corpus and toolset layer supplies scientific corpora, search and visit tools, retrieval operators and reading tools for metadata, abstracts and evidence chunks. The strategy–feedback loop between the Navigator and Librarians makes retrieval self-evolving: ranked evidence from each round updates the system's understanding of the user's intent and determines the next search direction, while the verified corpus layer keeps all ranked papers authentic and evidence-grounded.

1.2 PaSaMaster-Bench captures complex search intents

PaSaMaster-Bench evaluates whether retrieval systems can resolve the kinds of compositional search intents that arise in real research work [10, 18, 19]. Such intents are rarely reducible to a topic keyword: a user may simultaneously specify the scientific scope, required method, application domain, venue or time window, and exclusions that remove papers that look related but are not the intended target. Because these constraints are expressed in natural language rather than as explicit database filters, the system must infer the full intent before it can identify the correct paper set.

A representative task is: *Could you help me find empirical studies on federated learning for multi-hospital collaborative training of medical imaging AI models, published in Nature Medicine, Nature Communications or IEEE Transactions on Medical Imaging between 2023 and October 2025?* This is the kind of request a researcher might make when preparing a review, grant proposal or related-work section. It requires the system to satisfy the topic, method, application, venue, time and exclusion constraints jointly; a paper that only discusses federated learning, or only studies medical imaging, is not sufficient.

The benchmark contains 244 independent literature discovery tasks across 38 scientific disciplines. For each task, domain experts write the natural-language intent, decompose it into objective checklist items, assemble a broad candidate pool

through multiple retrieval channels and annotate every candidate against the checklist (Fig. 1b). The resulting target set is therefore stricter than topical relevance: a paper is correct only if it satisfies all required constraints. PaSaMaster-Bench measures whether a system can recover the intended paper set behind a realistic research question, not merely whether it can retrieve papers from the same field.

1.3 Experimental setup

We compare PaSaMaster with four groups of baselines. Lexical retrieval systems, represented by Google Scholar [1], search verified indexed records but mainly rely on keyword matching. Semantic retrieval systems, including OpenScholar [3] and Bohrium Science Navigator [4], use semantic matching or retrieval-augmented scientific search. Generative LLM baselines include DeepSeek, Kimi, MiniMax, GLM, Gemini and GPT [5–7, 21–23]. These LLMs are evaluated as tool-assisted search agents rather than closed-book models: each is equipped with Search and Visit tools and prompted to search, inspect evidence and verify sources before returning papers, following tool-use agent evaluation protocols [31–33]. For fixed-pipeline agentic retrieval, we use Google Scholar Labs [8] as a representative predefined agentic search workflow.

Retrieval quality is measured at $K = 20$ using standard information retrieval metrics [39, 40]. Recall@20 measures how many ground-truth target papers are recovered in the top 20 results; Precision@20 measures how many returned top-20 papers are genuine target papers; F1-score@20 summarizes the balance between recall and precision; and NDCG@20 measures whether relevant papers are ranked closer to the top. We also report per-query cost and source hallucination rate. Cost captures the computational expense of interpreting complex constraints and carrying out retrieval, while hallucination rate measures the proportion of returned papers that cannot be matched to a real scholarly record or contain provably wrong source metadata. Together, these metrics assess intent comprehension, source authenticity and cost-effective retrieval.

1.4 Accurate, hallucination-free retrieval at low cost

Table 2 reports the main results on PaSaMaster-Bench, and Fig. 2 summarizes cross-disciplinary robustness, source-error patterns and Ranker gains. Overall, PaSaMaster achieves the best retrieval quality while maintaining zero source hallucination and low computational cost. These results support the three central claims of our system: Recursive Self-Evolving retrieval improves understanding of complex natural-language search intents, intent–paper relevance ranking prevents hallucinated sources, and planning–retrieval separation enables cost-efficient large scale literature discovery.

Recursive Self-Evolving retrieval improves complex intent understanding. As shown in Table 2, PaSaMaster achieves the highest retrieval performance across all main quality metrics, with an NDCG@20 of 39.52, Recall@20 of 33.24, Precision@20 of 23.46, and F1-score@20 of 23.00. This demonstrates that PaSaMaster is better able to recover the target papers implied by complex multi-constraint research intents. Compared with Google Scholar [1], PaSaMaster improves F1-score@20 from 1.39 to 23.00, a 16.5× improvement, showing the severe limitation of keyword-centric

Table 2 Performance comparison of PaSaMaster against other methods. With zero hallucinations guaranteed, PaSaMaster achieves strong recall and ranking with low cost. The generative LLM baselines are allowed to call Search and Visit tools to find papers on the web before returning their final paper lists.

Method	NDCG	Recall	Precision	F1-score	Hallucination	Cost ($)
Lexical Retrieval Systems						
Google Scholar	2.07	1.69	1.48	1.39	0	–
Semantic Retrieval Systems						
OpenScholar	14.61	11.68	8.52	7.92	0	–
Bohrium Science Navigator	22.39	19.37	12.50	12.26	0	–
Generative LLMs						
DeepSeek-v3.2	35.82	24.76	15.35	15.56	12.94	0.28
Kimi-K2.5	37.80	28.08	16.95	17.36	26.59	0.16
MiniMax-M2.7	30.70	24.23	14.42	15.11	32.66	0.18
GLM-5	35.89	28.99	16.93	18.18	21.64	0.56
Gemini-3.1-pro	31.34	21.30	11.68	12.48	27.54	0.38
GPT-5.2	31.59	25.32	16.82	16.69	5.65	6.06
Fixed-Pipeline Agentic Retrieval						
Google Scholar Labs	30.54	29.01	18.79	18.87	0	–
Recursive Self-Evolving Agentic Retrieval						
PaSaMaster	**39.52**	**33.24**	**23.46**	**23.00**	**0**	**0.05**

retrieval under complex natural-language queries. Compared with semantic retrieval systems, PaSaMaster also substantially outperforms OpenScholar [3] and Bohrium Science Navigator [4], indicating that passive semantic matching is still insufficient for queries requiring constraint reasoning and intent refinement. Among generative LLMs, the strongest F1-score baseline is GLM-5 [23] with 18.18, while the fixed-pipeline agentic retrieval baseline Google Scholar Labs [8] achieves 18.87. PaSaMaster reaches 23.00, improving over these strongest baselines by 26.5% and 21.9%, respectively. Fig. 3 provides a mechanistic view of this advantage. In Fig. 3a, papers retrieved in successive rounds shift toward different topic regions, indicating that evidence from earlier rounds changes the system's interpretation of the query and opens new search directions. In Fig. 3b, the number of recovered ground-truth papers increases across rounds, showing that these new search directions lead to additional relevant papers rather than merely repeating the initial retrieval. Together, the two panels show that Recursive Self-Evolving retrieval improves complex intent understanding by using retrieved evidence to update cognition and guide later searches toward complementary parts of the intended paper set.

Intent–paper relevance ranking eliminates source hallucination. As shown in Table 2, PaSaMaster achieves 0% hallucination while maintaining the strongest retrieval quality. This result directly supports our design choice of treating literature discovery as intent–paper relevance ranking rather than generation. In contrast, the tool-assisted generative LLM baselines [5–7, 21–23] exhibit substantial hallucination rates, including 32.66% for MiniMax-M2.7, 27.54% for Gemini-3.1, 26.59% for Kimi-K2.5, 21.64% for GLM-5, 12.94% for DeepSeek-v3.2, and 5.65% for GPT-5.2. These

results show that even frontier LLMs with search and visit tools [28–33] remain vulnerable to fabricating or misreporting scientific sources [26, 27]. Fig. 2b further shows that hallucinations arise from multiple citation fields, including title, author, date, and link errors. By contrast, PaSaMaster ranks only papers retrieved from verified corpora and grounds relevance judgments in original paper evidence, thereby ensuring zero hallucination in source information.

Planning–retrieval separation reduces cost. Table 2 shows that PaSaMaster achieves this performance at a cost of only $0.05 per query. This is far below GPT-5.2 [5] at $6.06, GLM-5 [23] at $0.56, Gemini-3.1-pro [6] at $0.38, and DeepSeek-v3.2 [7] at $0.28. In particular, PaSaMaster outperforms GPT-5.2 in F1-score@20 by 37.8% while using only about 1% of its computational cost. This confirms the benefit of separating high-level planning from large-scale retrieval, enabling PaSaMaster to maintain high-quality retrieval at substantially lower cost.

Consistent gains across disciplines. Fig. 2 reports retrieval robustness, source hallucination and ranking performance across the 38 scientific disciplines in PaSaMaster-Bench. In Fig. 2a, PaSaMaster achieves leading F1-score distributions across multiple subject groups. Its performance range is shifted upward relative to competing paradigms, with a higher retrieval floor on difficult cases and a higher upper range on easier cases, indicating that the system improves both robustness and peak retrieval quality across disciplines. In Fig. 2c, the overall row shows consistent gains after Ranker training in NDCG@20, Recall@20 and Precision@20, while the subject rows show improvements across most disciplines. These gains reflect training on tens of thousands of multidisciplinary query–paper examples, which helps the Ranker generalize beyond a single domain and place relevant papers more accurately in the final list. Fig. 3b further shows that Recursive Self-Evolving retrieval expands the evidence space round by round across disciplines, increasing the average number of retrieved papers, high-score papers, and recovered ground-truth papers. Together, these results suggest that PaSaMaster's gains are not driven by a narrow domain-specific advantage. Instead, they reflect the generality of the three design principles: Recursive Self-Evolving retrieval supports complex intent understanding, evidence-grounded ranking ensures source authenticity, and planning–retrieval separation enables scalable retrieval across heterogeneous scientific domains.

2 Discussion

PaSaMaster shows that scientific literature discovery can be framed as a Recursive Self-Evolving, evidence-grounded ranking problem rather than as either keyword matching or citation generation. Across PaSaMaster-Bench, this formulation improves the recovery of target papers under complex search intents while maintaining zero source hallucination and substantially reducing computational cost. This is important because current systems face a structural trade-off: database-backed retrieval preserves source authenticity but often misses the intent behind a nuanced research need, whereas frontier LLMs can reason over richer requests but may fabricate or misreport scientific sources. PaSaMaster narrows this gap by keeping the output space restricted to verified papers while allowing the search process itself to evolve.

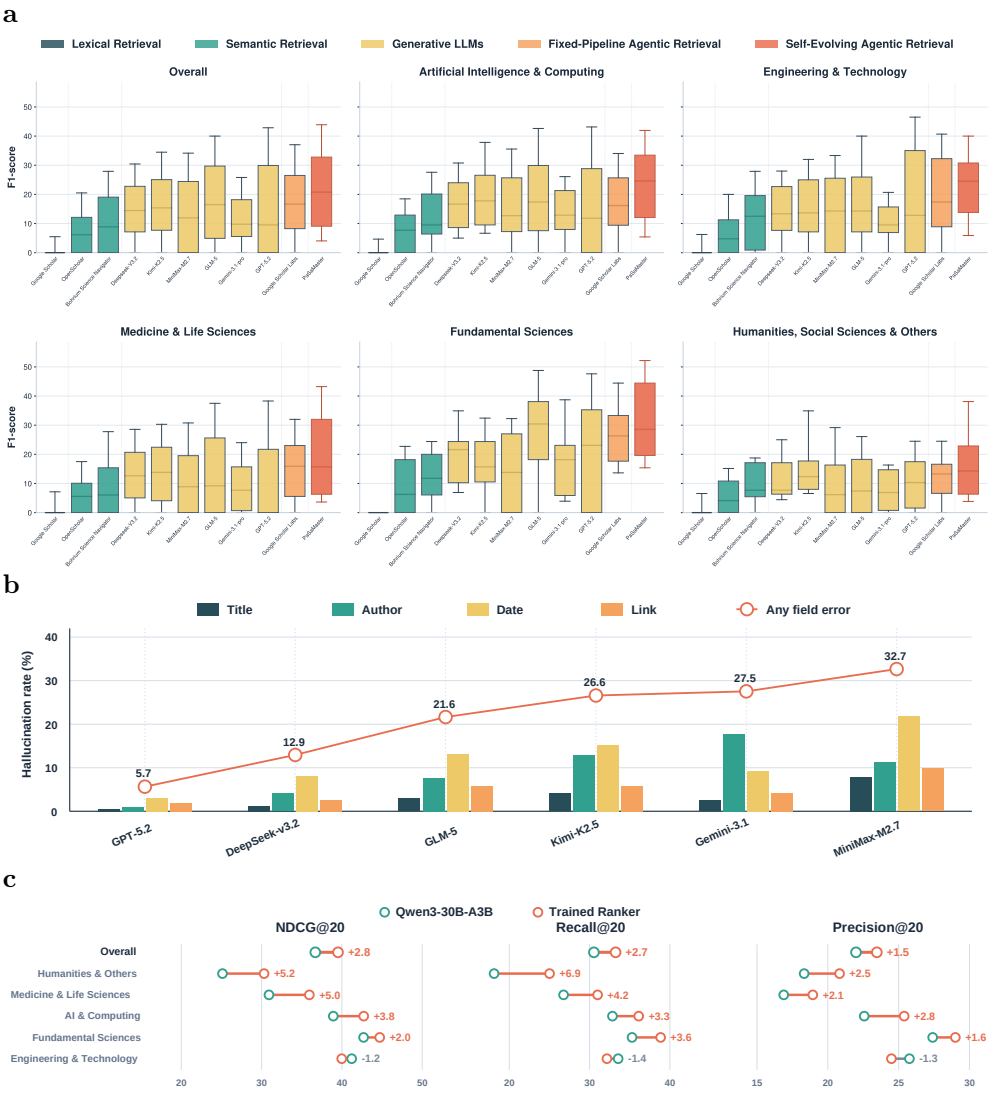

Fig. 2 Retrieval quality, source hallucination and ranking performance on PaSaMaster-Bench. a, F1-score distributions comparing different retrieval systems across scientific disciplines. PaSaMaster achieves the strongest overall retrieval performance and leads broadly across subject areas, with both a higher lower bound and a higher upper performance range than competing paradigms. **b,** Citation-field hallucination rates of tool-assisted generative LLMs, ordered by overall source hallucination rate. Bars show field-specific title, author, date and link errors, and the line marks the overall proportion of returned papers with any source error. **c,** Overall and per-discipline before–after ranking performance after Ranker training, measured by NDCG@20, Recall@20 and Precision@20. The overall gain is shown first, followed by subject groups ordered by average gain, highlighting broad improvements after multidisciplinary Ranker training.

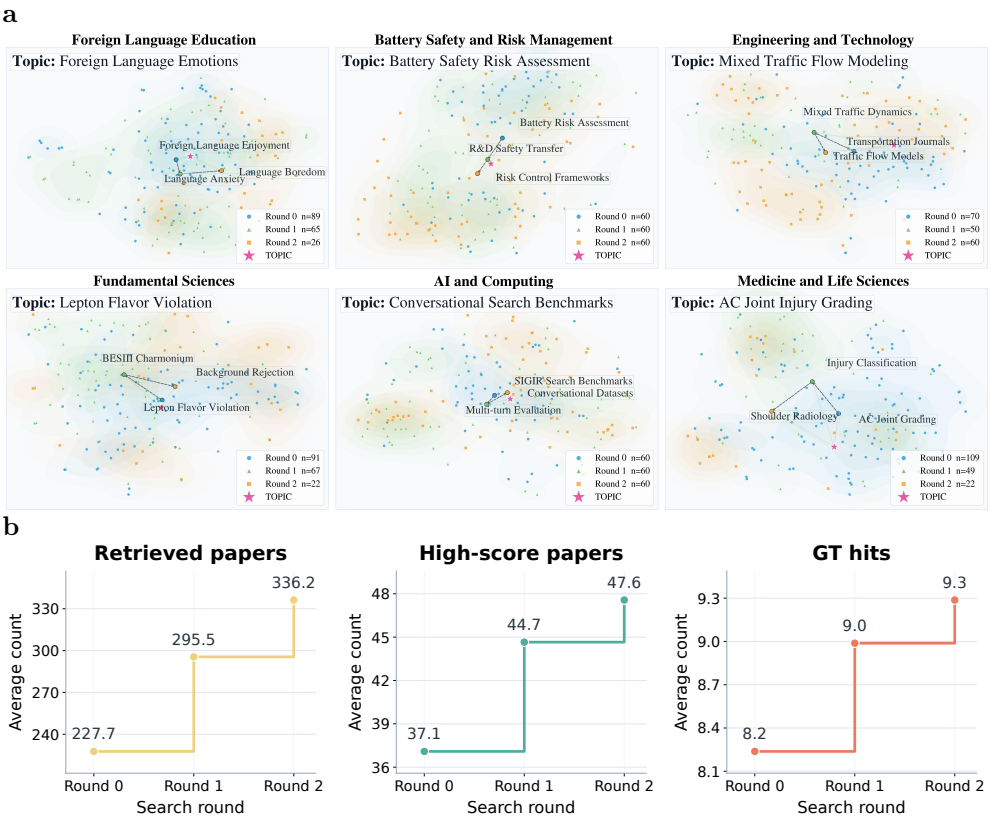

Fig. 3 Evidence feedback drives intent evolution and target-paper discovery. a, t-SNE visualizations of six representative retrieval cases across successive search rounds. Each color denotes papers retrieved in one round, the star marks the original query, density shading indicates retrieved-paper concentration and annotations identify dominant topics. Across rounds, retrieved papers shift toward different topic regions and form distinct distributions, showing that PaSaMaster develops new understanding of the user intent from previously retrieved evidence and uses it to explore new search directions. This evolving cognition helps later rounds move beyond the initial query interpretation and retrieve complementary relevant papers. **b,** Retrieval dynamics averaged over all disciplines. Retrieved papers from each round are fed back to the Navigator to refine its understanding of the user intent and update the next search strategy. As the self-evolving process proceeds, PaSaMaster retrieves a larger candidate pool, identifies more high-score papers and recovers more ground-truth target papers. The increase in ground-truth hits across rounds shows that each iteration contributes additional relevant papers rather than merely repeating earlier retrieval results.

The central mechanism is to use ranked evidence to continually update the system's understanding of the user's intent during retrieval. This makes PaSaMaster Recursive Self-Evolving: the search process does not merely execute an initial query, but progressively revises what the query means as evidence accumulates. Researchers rarely know the exact best query before seeing the literature; they search, inspect partial results, recognize missing constraints and refine their direction. PaSaMaster operationalizes this process by letting the Navigator revise the retrieval strategy and verification checklist after observing scored candidates from the Librarian swarm. The resulting

loop helps the system move beyond a static interpretation of the initial query, which is especially valuable for multi-constraint intents in which relevance depends on combinations of topic, method, dataset, application context and exclusion criteria. The cross-disciplinary results suggest that this mechanism is not only a domain-specific optimization, but a general strategy for aligning retrieval with complex scientific needs.

Equally important is the decision to rank papers rather than generate citations. In many LLM-based literature workflows, hallucination arises because the model is asked to produce bibliographic objects directly from parametric memory or from partially grounded context. PaSaMaster changes this failure mode by requiring every candidate to be retrieved from a verified corpus and every relevance judgment to be supported by evidence from the original paper. The system therefore does not promise that every relevance judgment is perfect, but it does make each relevance score traceable to paper-level evidence and ensures that recommended sources are real and auditable. This distinction matters for scientific use because fabricated papers can distort a literature review, create false evidence chains and mislead subsequent hypothesis formation or experiment design. PaSaMaster reduces this risk by making each recommendation traceable to a real paper and to the evidence used to judge its relevance.

The low-cost design also shapes the potential use of PaSaMaster. Literature discovery is not a one-off task; researchers repeatedly search while designing projects, writing related work, checking novelty and updating reviews. A system that relies on frontier LLMs for every retrieval and scoring operation is difficult to deploy at this frequency and scale. By separating high-level planning from large-scale retrieval and lightweight relevance scoring, PaSaMaster makes agentic literature discovery more practical for broad, multidisciplinary use. In this role, the system should be viewed as an assistive layer that expands and organizes the candidate evidence space, not as a replacement for expert judgment. Its value is strongest when it delivers more comprehensive and accurate retrieval at low cost, exposes why papers were recommended and makes omissions easier to diagnose.

Several limitations remain. First, PaSaMaster-Bench is expert-curated, and its target sets may still be influenced by the disciplinary expertise, prior knowledge and search horizon of the annotators. We mitigated this limitation by using multiple retrieval channels to help experts assemble broad candidate pools, and by requiring elementwise checklist scoring so that each candidate paper is verified against the stated intent rather than accepted by impression alone. Second, the current evaluation focuses on top-ranked paper retrieval rather than the quality of downstream literature synthesis, hypothesis generation or manuscript writing. Future work should therefore test whether improved retrieval leads to better scientific outputs in human-in-the-loop settings, including expert assessments of coverage, novelty, usefulness and trustworthiness. Third, PaSaMaster could be extended from retrieving relevant papers to helping scientists reconstruct the historical development of a field, identify emerging research trajectories and generate candidate research directions grounded in the literature. Such capabilities would help researchers rapidly understand how a topic has evolved and use verified evidence to define new research questions. Finally, although PaSaMaster eliminates source hallucination by construction, relevance scoring can still inherit biases from corpora, models and checklist design. A deployment-ready system

should therefore expose evidence, uncertainty and failure modes clearly enough for researchers to audit its recommendations. Taken together, these results indicate that self-evolving, verified retrieval is a promising foundation for trustworthy AI-assisted science, but its full value will depend on transparent evaluation, continual corpus updating and careful integration into expert research workflows.

3 Methods

3.1 Overview of PaSaMaster

PaSaMaster is an agentic Recursive Self-Evolving literature retrieval system that maps a complex natural-language search intent q to a ranked, evidence-grounded paper set $\mathcal{P} = [p_1, p_2, \ldots, p_k]$, where p_i is the ith recommended paper and k is the output list length. Its design follows three principles that directly address the limitations of existing literature retrieval paradigms: *Recursive Self-Evolving retrieval*, *hallucination-free intent–paper relevance ranking*, and *cost-efficient planning–retrieval separation*. Rather than generating paper lists from parametric memory, PaSaMaster retrieves real papers from customized scientific corpora, verifies their relevance using original evidence, and iteratively refines the search intent based on ranked retrieval results.

Formally, PaSaMaster operates over a customized scientific corpus $\mathcal{D}$ and an agent-accessible operator toolset $\mathcal{T}$. Given query q, a Navigator with policy π_{Nav} first produces a retrieval strategy S and a query-specific verification checklist $C = [c_1, c_2, \ldots, c_m]$, where m is the number of checklist items and each checkpoint c_j encodes one concrete requirement that a relevant paper must satisfy. The system also uses N parallel Librarian agents, denoted by $\{\pi_{\text{Lib}}^{(i)}\}_{i=1}^{N}$, where $\pi_{\text{Lib}}^{(i)}$ is the policy of the ith Librarian. Let PLAN denote Navigator planning, RETRIEVE denote corpus search, VERIFY denote evidence-grounded candidate scoring and RERANK denote final listwise ordering. The overall retrieval pipeline is:

$$\langle S,\, C\rangle = \textsc{Plan}(q, \pi_{\text{Nav}}), \tag{1}$$

$$\mathcal{P}_{\text{init}} = \textsc{Retrieve}\Big(S; \mathcal{D}, \mathcal{T}, \{\pi_{\text{Lib}}^{(i)}\}_{i=1}^{N}\Big), \tag{2}$$

$$\mathcal{P}_{\text{scored}} = \textsc{Verify}\Big(\mathcal{P}_{\text{init}}, C; \mathcal{D}, \mathcal{T}, \{\pi_{\text{Lib}}^{(i)}\}_{i=1}^{N}\Big), \tag{3}$$

$$\mathcal{P} = \textsc{Rerank}(\mathcal{P}_{\text{scored}}), \tag{4}$$

Here $\mathcal{P}_{\text{init}}$ is the initially retrieved candidate set, $\mathcal{P}_{\text{scored}}$ is the evidence-scored candidate set and $\mathcal{P}$ is the final ranked paper list. Equations 1–4 summarize PaSaMaster as a staged process in which planning defines the search, Librarians retrieve and verify papers from trusted tools, and reranking converts scored candidates into the final recommendation list.

3.1.1 Recursive Self-Evolving Retrieval from Ranked Evidence

The first core design of PaSaMaster is to transform literature retrieval from one-shot query–document matching into a Recursive Self-Evolving search process. Existing retrieval systems typically fix their interpretation of the user query at the beginning [1, 3, 4, 8, 9] and execute retrieval under this static understanding. PaSaMaster instead treats retrieval as an iterative process in which ranked evidence is used to update the system's understanding of the research intent.

The process is coordinated by the Navigator agent. Given the initial query q, the Navigator first analyzes the user's research intent and generates two outputs: a retrieval strategy S, specifying what should be searched, and a verification checklist C, specifying how candidate papers should be judged. Let t index the retrieval round, and let $S^{(t)}$, $C^{(t)}$ and $\mathcal{P}^{(t)}_{\text{scored}}$ denote the retrieval strategy, checklist and scored candidate set at round t. After each retrieval round, the Navigator inspects the ranked results, identifies missing coverage, ambiguous constraints, or under-explored directions, and refines the strategy and checklist for the next round through the reflection operator Reflect:

$$\langle S^{(t+1)}, C^{(t+1)} \rangle = \textsc{Reflect}\Big(q, S^{(t)}, C^{(t)}, \mathcal{P}^{(t)}_{\text{scored}}\Big), \tag{5}$$

Equation 5 formalizes the self-evolving step: ranked evidence from round t updates the system's understanding of the query and produces the strategy and checklist used in round $t+1$.

3.1.2 Hallucination-Free Intent–Paper Relevance Ranking

The second core design is to prevent hallucinated sources by formulating literature discovery as intent–paper relevance ranking rather than generation. PaSaMaster never asks an LLM to synthesize citations or paper lists directly from parametric memory. Instead, every candidate paper must be retrieved from a verified scientific corpus $\mathcal{D}$, and every relevance judgment must be grounded in traceable evidence from the original paper.

To support verifiable retrieval and evidence grounding, PaSaMaster restructures over 160 million papers into a three-tier agent-native repository [41]. The repository contains $\mathcal{D}^{\text{meta}}$ for structured metadata, $\mathcal{D}^{\text{abs}}$ for abstract-level representations used in coarse semantic filtering and $\mathcal{D}^{\text{chunk}}$ for passage-level evidence chunks segmented from full texts. The corpus is represented as:

$$\mathcal{D} = \{\mathcal{D}^{\text{meta}}, \mathcal{D}^{\text{abs}}, \mathcal{D}^{\text{chunk}}\}, \tag{6}$$

Equation 6 defines the corpus layout used to keep metadata retrieval, abstract-level filtering and passage-level evidence grounding separate but jointly accessible to the agents.

For each candidate paper p and checklist item c_j, let $\mathcal{D}^{\text{chunk}}_p$ denote the set of evidence chunks belonging to paper p, let e denote one candidate chunk in this set, let $\phi(\cdot)$ be the shared text encoder, let ℓ denote the number of evidence chunks to retrieve, let Top_ℓ denote selection of the ℓ highest-scoring chunks and let $\mathcal{E}_p(c_j)$ denote the top-ℓ evidence chunks selected to support the judgment for checklist item c_j. The

Evidence Chunk Locator retrieves supporting passages by cosine similarity:

$$\mathcal{E}_p(c_j) = \underset{\ell,\, e \in \mathcal{D}_p^{\text{chunk}}}{\text{Top}} \cos(\phi(c_j), \phi(e)) \,, \tag{7}$$

Equation 7 binds each checklist judgment to explicit textual evidence by selecting the passages in a paper that are most semantically aligned with the requirement being checked.

Each candidate paper is then evaluated by a lightweight Scorer model trained through a multidisciplinary distillation pipeline. We first sample seed topics across disciplines and use them to construct initial literature queries and seed paper collections. The retrieved papers are then clustered to obtain finer-grained topic groups, from which we synthesize realistic search queries and query-specific checklists covering topical, methodological, application metadata constraints and exclusion criteria. Each synthesized query is run through the PaSaMaster retrieval pipeline to obtain candidate papers, producing multidisciplinary query–paper pairs that include positive matches, partial matches and hard negatives. A stronger teacher model then labels each pair according to the checklist, assigning checkpoint-level scores, evidence-grounded rationales and holistic relevance judgments.

This training process gives the lightweight Scorer broad, expert-style relevance assessment ability across disciplines. At inference time, for every paper p and checkpoint c_j, the Scorer outputs a satisfaction score $s_j(p) \in \{1, 2, 3, 4, 5\}$ and an evidence-grounded rationale, where larger values indicate stronger satisfaction of the checklist item. For a paper p, let $\bar{s}(p)$ denote the average checklist satisfaction score over the m checklist items:

$$\bar{s}(p) = \frac{1}{m} \sum_{j=1}^{m} s_j(p). \tag{8}$$

Equation 8 summarizes checkpoint-level evidence into a single criterion-level relevance signal for candidate paper p.

To incorporate holistic confidence, PaSaMaster also extracts the Scorer model's calibrated output probability $\rho \in [0, 1]$ for its overall relevance judgment. Let $\mathcal{S}(p)$ denote the final normalized relevance score for paper p. The final relevance score is:

$$\mathcal{S}(p) = \frac{\bar{s}(p) + \rho}{6}, \tag{9}$$

where the denominator 6 normalizes the maximum possible value of $\bar{s}(p) + \rho$, because $\bar{s}(p) \leq 5$ and $\rho \leq 1$. Equation 9 combines checklist satisfaction and holistic confidence into one paper-level relevance score. The top candidates are then passed to a listwise reranker for global cross-paper comparison. The final result $\mathcal{P}$ is therefore a relevance-ranked list of real papers, with each recommendation traceable to paper-level evidence.

3.1.3 Cost-Efficient Planning–Retrieval Separation

The third core design is planning–retrieval separation, which improves scalability by using frontier LLMs only where they are most valuable. Frontier LLMs are effective for

understanding, decomposing, and refining complex research intents, but using them for every retrieval, reading, and ranking operation would be unnecessarily expensive. PaSaMaster therefore assigns high-level reasoning to the Navigator and delegates large-scale retrieval and relevance scoring to customized corpora, and lightweight parallel Librarian agents.

Let $\mathcal{T}^{\text{ret}}$ denote the retrieval tools used to construct candidate pools, and let $\mathcal{T}^{\text{read}}$ denote the reading tools used to inspect metadata, abstracts and evidence chunks. The operator toolset is divided into these two subsets:

$$\mathcal{T} = \mathcal{T}^{\text{ret}} \cup \mathcal{T}^{\text{read}}. \tag{10}$$

Equation 10 states that PaSaMaster separates tools for finding candidate papers from tools for reading and verifying them, which supports cost-efficient division of labor.

The retrieval tools $\mathcal{T}^{\text{ret}}$ construct a broad candidate pool through complementary retrieval channels, including Semantic Direct Retrieval, Citation Network Expansion, and Web-to-Repository Verification. Let o denote one retrieval operator in $\mathcal{T}^{\text{ret}}$, let $\textsc{Retrieve}_o(S, \mathcal{D})$ denote the candidates returned by operator o under strategy S over corpus $\mathcal{D}$, and let $\mathcal{C}_{\text{init}}$ denote the union of candidates returned by all retrieval operators:

$$\mathcal{C}_{\text{init}} = \bigcup_{o \in \mathcal{T}^{\text{ret}}} \textsc{Retrieve}_o(S, \mathcal{D}). \tag{11}$$

Equation 11 defines the initial candidate pool as the union of complementary retrieval channels, increasing coverage before evidence verification and reranking. Semantic Direct Retrieval provides high-precision semantic candidates, Citation Network Expansion follows citation links to surface structurally related papers, and Web-to-Repository Verification maps external web findings back to verified repository entries. The reading tools $\mathcal{T}^{\text{read}}$ then support efficient metadata lookup, abstract reading, and evidence-chunk localization, avoiding expensive full-document reading and substantially reducing computational cost.

Finally, the distilled Scorer serves as the verification component of each Librarian agent. Given a query-specific checklist and retrieved evidence chunks, it assigns checklist-level scores, generates evidence-grounded rationales and produces a holistic relevance judgment without repeatedly invoking a frontier LLM for every candidate paper. This enables Librarian agents to reproduce expert-style structured verification at much lower inference cost over large scientific corpora.

3.2 PaSaMaster-Bench

PaSaMaster-Bench is designed to evaluate literature retrieval systems under conditions that resemble real scientific paper discovery. In practice, researchers rarely ask only for a topic; they ask natural-language questions that combine scientific scope, methods, application setting, metadata restrictions and exclusions, often while searching across the open web rather than a fixed corpus. A benchmark that omits any of these dimensions can overestimate retrieval ability: short keyword-like queries understate intent reasoning, loose relevance labels do not test full constraint satisfaction,

Table 3 Comparison with existing literature search and research-agent benchmarks. The table compares our benchmark with representative benchmarks across four properties required for realistic scientific paper discovery, including natural-language queries, complex compositional search needs, multidisciplinary coverage and real-web search.

Benchmark Dimension	RealScholar Query [9]	Research Arena [36]	AstaBench Paper Finder [37]	AutoResearch Bench [38]	LitSearch [19]	PaSaMaster-Bench
Natural-language queries	✓	✓	✓	✓	✓	✓
Complex search intent	×	×	×	✓	✓	✓
Multidisciplinary coverage	×	✓	✓	×	×	✓
Search in a real web environment	✓	×	×	×	×	✓

single-domain tasks hide cross-disciplinary fragility, and fixed-corpus settings bypass source verification in real search environments.

Table 3 summarizes this gap. Existing literature-search and research-agent benchmarks [9, 19, 36–38] cover useful pieces of the problem, but typically miss at least one property needed for realistic paper discovery: complex compositional natural-language intents, broad multidisciplinary coverage, or real-web search. PaSaMaster-Bench is constructed to satisfy all four properties simultaneously. It contains **244 independent tasks** across **38 scientific disciplines**, and each task pairs a realistic natural-language search intent with an expert-annotated target paper set $\mathcal{P}^*$.

This design makes PaSaMaster-Bench a direct testbed for the capabilities that PaSaMaster is built to provide. A system must infer the user's complete intent, search in realistic environments, return authentic papers and filter candidates by paper-level evidence. A paper is counted as correct only if it satisfies all expert-defined checklist criteria, rather than merely sharing the same topic. The benchmark therefore evaluates whether a retrieval system can recover the intended paper set behind a real research need, not simply whether it can retrieve plausible or broadly relevant papers.

3.2.1 Data Curation

The curation pipeline is built to preserve realism while making evaluation objective (Fig. 1b). First, domain experts write search intents grounded in authentic research scenarios across the 38 disciplines. The intent is kept in natural language so that the task resembles how a scientist would ask for papers, but it is also decomposed into a checklist of objective criteria. These criteria specify the topical scope, required methods, application setting, dataset or benchmark conditions, publication restrictions and exclusion rules that define the target paper set.

Second, each query is executed through multiple retrieval channels to approximate a real open-web search process. We use web-enabled frontier LLMs, PaSaMaster's native search engine and traditional web search to build an intentionally broad candidate pool. The purpose is not to treat any retriever as ground truth, but to expose experts to a diverse set of possible targets, including papers that may be missed by one search channel. Retrieved papers are verified, deduplicated and mapped into a unified candidate set before annotation.

Third, domain experts annotate candidates against the checklist item by item. A paper is admitted into $\mathcal{P}^*$ only when it satisfies every required criterion. Papers that are topically related but fail a method requirement, application setting, metadata constraint or exclusion rule are marked as incorrect. This elementwise annotation converts complex natural-language needs into verifiable target sets while preserving the compositional difficulty of the original query.

3.2.2 Evaluation Protocol

The evaluation protocol asks whether a system can reconstruct the target paper set implied by a complex user intent. Given a query, the system must autonomously search, verify and return a ranked list $\mathcal{P}_{\text{agent}} = [p_1, p_2, \ldots, p_k]$, where p_i is the ith returned paper. The list is compared with $\mathcal{P}^*$, the expert-annotated set of papers satisfying all checklist criteria. This setup makes the evaluation stricter than topical relevance: a returned paper is useful only if it satisfies the complete intent.

Coverage and ranking quality are evaluated with standard ranking metrics. At cutoff $K = 20$, Recall@K measures whether the system can cover the intended paper set; Precision@K measures whether returned papers satisfy the expert checklist; F1@K captures the balance between coverage and constraint satisfaction; and NDCG@K measures whether the most relevant target papers are ranked near the top [39, 40, 42]. In this benchmark, these metrics jointly test natural-language intent comprehension, paper-level filtering and ranking under compositional constraints.

Beyond retrieval quality, PaSaMaster-Bench evaluates source authenticity and cost efficiency, two requirements for deployable scientific search systems. Source hallucination rate measures the proportion of returned papers that cannot be matched to a real scholarly record or contain provably wrong source metadata. Token usage and per-query cost measure whether complex retrieval can remain low-overhead enough to scale across repeated, broad use in scientific workflows. For generative LLM baselines, we evaluate tool-assisted literature search rather than direct parametric answering: each model is equipped with Search and Visit tools and prompted to perform ReAct-style search, evidence inspection and source verification before producing a final ranked list [28–33]. This setting tests whether LLM agents can use external tools to discover real, constraint-satisfying papers rather than merely generate plausible citations.

Thus, PaSaMaster-Bench measures four capabilities required for realistic scientific literature discovery: *intent comprehension*, *constraint-satisfying retrieval*, *ranking quality* and *source authenticity*. These are also the capabilities targeted by PaSaMaster's self-evolving retrieval, evidence-grounded ranking and planning–retrieval separation.

3.2.3 Search Intent Examples

PaSaMaster-Bench is designed around natural-language intents that mirror how researchers actually search for literature. The queries cover four common research needs in scientific work: finding a known type of study, locating target papers when only the research problem is clear, preparing a literature review with strict metadata constraints such as venue or time period, and avoiding superficially similar papers that

do not meet the actual requirements. Together, these scenarios make the benchmark closer to real literature discovery than keyword-style retrieval tasks. The examples below show how PaSaMaster-Bench combines topical scope, methodological requirements, application context, metadata restrictions and exclusion criteria, requiring systems to recover papers that satisfy the complete intent rather than merely share keywords.

Example query 1: *I need papers on identifying RNA modifications using mass spectrometry that employ machine learning algorithms for spectrum annotation. The mass spectrometry approach should use tandem MS with fragmentation patterns.*

This is a direct intent query. The user knows what they want to find, but the target is still compositional: relevant papers must jointly concern RNA modification identification, mass spectrometry, machine-learning-based spectrum annotation and tandem-MS fragmentation evidence.

Example query 2: *I'm preparing to construct a computational prediction system for regioselectivity of transition metal catalyzed cross-coupling on mutihalogen reactants, but do not know which prediciton model/descriptors should I include in our method. Can you recommend papers with methods to predict such regioselectivity? Do a thorough search for these papers, provide as many papers as you can find.*

This is a problem-driven query. The user does not know the exact model family, descriptors or keywords to search for, but clearly states a research bottleneck. A capable retrieval agent must infer the relevant methodological space and translate the user's need into search criteria before recommending papers.

Example query 3: *Could you help me find empirical studies on Federated Learning for multi-hospital collaborative training of medical imaging AI models, published in Nature Medicine, Nature Communications, or IEEE Transactions on Medical Imaging between 2023 and October 2025?*

This is a metadata-constrained query. The system must retrieve papers on the target topic while also enforcing publication type and time range. It therefore tests whether retrieval can combine semantic relevance with strict paper-level metadata constraints.

Example query 4: *Can you help check if there is a paper that has designed a gel electrolyte for batteries, where this gel electrolyte is not for lithium-ion batteries but for lithium-sulfur batteries? This gel electrolyte is not for other systems, but specifically for the PDOL system.*

This exclusion-sensitive query requires selecting a specific battery system and electrolyte chemistry while rejecting nearby but incorrect targets, such as lithium-ion batteries or non-PDOL gel electrolytes. The task therefore requires judgment over both positive and negative conditions, not just semantic similarity to the topic.

Together, these examples cover much of the search behavior encountered in real scientific work: directly specified retrieval, exploratory recommendation from a research problem, metadata-restricted paper finding and fine-grained exclusion of false positives. PaSaMaster-Bench therefore evaluates both the realism and the compositional difficulty of literature discovery, requiring systems to understand what the user means, search vertically within a domain and verify whether each candidate paper truly satisfies the intended requirements.

References

[1] Google: Google Scholar. https://scholar.google.com/ (2026)

[2] National Center for Biotechnology Information: PubMed. https://pubmed.ncbi.nlm.nih.gov/ Accessed 2026-05-08

[3] Asai, A., He, J., Shao, R., Shi, W., Singh, A., Chang, J.C., Lo, K., Soldaini, L., Feldman, S., D'arcy, M., Wadden, D., Latzke, M., Tian, M., Ji, P., Liu, S., Tong, H., Wu, B., Xiong, Y., Zettlemoyer, L., Neubig, G., Weld, D., Downey, D., Yih, W.-t., Koh, P.W., Hajishirzi, H.: OpenScholar: Synthesizing Scientific Literature with Retrieval-augmented LMs (2024). http://arxiv.org/abs/2411.14199

[4] Zhang, L., Chen, S., Cai, Y., Chai, J., Chang, J., Chen, K., Chen, Z.X., Ding, Z., Du, Y., Gao, Y., Gao, Y., Gao, J., Gao, Z., Gu, Q., Hong, Y., Huang, Y., Fang, X., Ji, X., Ke, G., Lei, Z., Li, X., Li, Y., Liao, R., Lin, H., Lin, X., Liu, Y., Liu, X., Liu, Z., Lu, J., Miao, T., Que, H., Sun, W., Wang, Y., Wu, B., Xue, T., Ye, R., Zeng, J., Zhang, D., Zhang, J., Zhang, L., Zhang, T., Zhang, W., Zhang, Y., Zhang, Z., Zheng, H., Zhou, H., Zhu, T., Zhu, X., Zhou, Q., E, W.: Bohrium + SciMaster: Building the Infrastructure and Ecosystem for Agentic Science at Scale (2025). https://arxiv.org/abs/2512.20469

[5] OpenAI: Introducing GPT-5.4 (2026). https://openai.com/zh-Hans-CN/index/introducing-gpt-5-4/

[6] Google DeepMind: Gemini 3.1 Pro: A smarter model for your most complex tasks (2026). https://blog.google/innovation-and-ai/models-and-research/gemini-models/gemini-3-1-pro/

[7] DeepSeek-AI, Liu, A., Mei, A., Lin, B., Xue, B., Wang, B., Xu, B., Wu, B., Zhang, B., Lin, C., Dong, C., Lu, C., Zhao, C., Deng, C., Xu, C., Ruan, C., Dai, D., Guo, D., Yang, D., Chen, D., Li, E., Zhou, F., Lin, F., Dai, F., Hao, G., Chen, G., Li, G., Zhang, H., Xu, H., Li, H., Liang, H., Wei, H., Zhang, H., Luo, H., Ji, H., Ding, H., Tang, H., Cao, H., Gao, H., Qu, H., Zeng, H., Huang, J., Li, J., Xu, J., Hu, J., Chen, J., Xiang, J., Yuan, J., Cheng, J., Zhu, J., Ran, J., Jiang, J., Qiu, J., Li, J., Song, J., Dong, K., Gao, K., Guan, K., Huang, K., Zhou, K., Huang, K., Yu, K., Wang, L., Zhang, L., Wang, L., Zhao, L., Yin, L., Guo, L., Luo, L., Ma, L., Wang, L., Zhang, L., Zhang, M., Zhang, M., Tang, M., Zhou, M., Huang, P., Cong, P., Wang, P., Wang, Q., Zhu, Q., Li, Q., Chen, Q., Du, Q., Xu, R.,

Ge, R., Zhang, R., Pan, R., Wang, R., Yin, R., Xu, R., Shen, R., Zhang, R., Lu, S., Zhou, S., Chen, S., Cai, S., Chen, S., Hu, S., Liu, S., Hu, S., Ma, S., Wang, S., Yu, S., Zhou, S., Pan, S., Zhou, S., Ni, T., Yun, T., Pei, T., Ye, T., Yue, T., Zeng, W., Liu, W., Liang, W., Gao, W., Zhang, W., Bi, X., Liu, X., Wang, X., Chen, X., Zhang, X., Nie, X., Cheng, X., Liu, X., Xie, X., Liu, X., Yu, X., Yang, X., Li, X., Su, X., Lin, X., Zhang, Y., Xu, Y., Li, Y., Zhao, Y., Sun, Y., Wang, Y., Yu, Y., Zhang, Y., Shi, Y., Xiong, Y., He, Y., Piao, Y., Wang, Y., Tan, Y., Ma, Y., Liu, Y., Guo, Y., Wu, Y., Ou, Y., Wang, Y., Gong, Y., Zou, Y., Xiong, Y., Luo, Y., You, Y., Liu, Y., Zhou, Y., Ren, Z., Sha, Z., Fu, Z., Xu, Z., Xie, Z., Zhang, Z., Hao, Z., Gou, Z., Ma, Z., Yan, Z., Shao, Z., Wu, Z., Li, Z., Gu, Z., Zhu, Z., Li, Z., Xie, Z., Gao, Z., Pan, Z.: DeepSeek-V3.2: Pushing the Frontier of Open Large Language Models (2025). https://arxiv.org/abs/2512.02556

[8] Google: Scholar Labs: An AI Powered Scholar Search. Google Scholar Blog (2025). https://scholar.googleblog.com/2025/11/scholar-labs-ai-powered-scholar-search.html

[9] He, Y., Huang, G., Feng, P., Lin, Y., Zhang, Y., Li, H., E, W.: PaSa: An LLM Agent for Comprehensive Academic Paper Search (2025). https://arxiv.org/abs/2501.10120

[10] Gusenbauer, M., Haddaway, N.R.: What every researcher should know about searching—clarified concepts, search advice, and an agenda to improve finding in academia. Research Synthesis Methods **12**(2), 136–147 (2021) https://doi.org/10.1002/jrsm.1457

[11] Fortunato, S., Bergstrom, C.T., Börner, K., Evans, J.A., Helbing, D., Milojević, S., Petersen, A.M., Radicchi, F., Sinatra, R., Uzzi, B., Vespignani, A., Waltman, L., Wang, D., Barabási, A.-L.: Science of science. Science **359**(6379), 0185 (2018) https://doi.org/10.1126/science.aao0185

[12] Bornmann, L., Mutz, R.: Growth rates of modern science: A bibliometric analysis based on the number of publications and cited references (2014) arXiv:1402.4578 [cs.DL]

[13] Landhuis, E.: Scientific literature: Information overload. Nature **535**(7612), 457–458 (2016)

[14] Houssard, A., Gargiulo, F., Venturini, T., Tubaro, P., Di Bona, G.: The Gerontocratization of Science: How Hypergrowth Reshapes Knowledge Circulation (2025). https://arxiv.org/abs/2410.00788

[15] Wang, H., Fu, T., Du, Y., Gao, W., Huang, K., Liu, Z., Chandak, P., Liu, S., Van Katwyk, P., Deac, A., Anandkumar, A., Bergen, K., Gomes, C.P., Ho, S., Kohli, P., Lasenby, J., Leskovec, J., Liu, T.-Y., Manrai, A., Marks, D., Ramsundar, B., Song, L., Sun, J., Tang, J., Veličković, P., Welling, M., Zhang, L., Coley, C.W., Bengio, Y., Zitnik, M.: Scientific discovery in the

age of artificial intelligence. Nature **620**, 47–60 (2023) https://doi.org/10.1038/s41586-023-06221-2

[16] Furnas, G.W., Landauer, T.K., Gomez, L.M., Dumais, S.T.: The vocabulary problem in human-system communication. Communications of the ACM **30**(11), 964–971 (1987) https://doi.org/10.1145/32206.32212

[17] Beel, J., Gipp, B., Wilde, E.: Academic search engine optimization (ASEO): Optimizing scholarly literature for Google Scholar and co. Journal of Scholarly Publishing **41**(2), 176–190 (2010) https://doi.org/10.3138/jsp.41.2.176

[18] White, R.W., Roth, R.A.: Exploratory Search: Beyond the Query-Response Paradigm. Morgan & Claypool Publishers, San Rafael, CA (2009). https://doi.org/10.2200/S00174ED1V01Y200901ICR003

[19] Ajith, A., Xia, M., Chevalier, A., Goyal, T., Chen, D., Gao, T.: LitSearch: A Retrieval Benchmark for Scientific Literature Search (2024). https://arxiv.org/abs/2407.18940

[20] Zhang, Y., Khan, S.A., Mahmud, A., Yang, H., Lavin, A., Levin, M., Frey, J.G., Dunnmon, J., Evans, J., Bundy, A., Džeroski, S., Tegnér, J., Zenil, H.: Exploring the role of large language models in the scientific method: from hypothesis to discovery. npj Artificial Intelligence **1**(1), 14 (2025) https://doi.org/10.1038/s44387-025-00019-5

[21] Team, K., Bai, Y., Bao, Y., Charles, Y., Chen, C., Chen, G., Chen, H., Chen, H., Chen, J., Chen, N., Chen, R., Chen, Y., Chen, Y., Chen, Y., Chen, Z., Cui, J., Ding, H., Dong, M., Du, A., Du, C., Du, D., Du, Y., Fan, Y., Feng, Y., Fu, K., Gao, B., Gao, C., Gao, H., Gao, P., Gao, T., Ge, Y., Geng, S., Gu, Q., Gu, X., Guan, L., Guo, H., Guo, J., Hao, X., He, T., He, W., He, W., He, Y., Hong, C., Hu, H., Hu, Y., Hu, Z., Huang, W., Huang, Z., Huang, Z., Jiang, T., Jiang, Z., Jin, X., Kang, Y., Lai, G., Li, C., Li, F., Li, H., Li, M., Li, W., Li, Y., Li, Y., Li, Y., Li, Z., Li, Z., Lin, H., Lin, X., Lin, Z., Liu, C., Liu, C., Liu, H., Liu, J., Liu, J., Liu, L., Liu, S., Liu, T., Liu, W., Liu, Y., Liu, Y., Liu, Y., Liu, Y., Liu, Z., Lu, E., Lu, H., Lu, L., Luo, Y., Ma, S., Ma, X., Ma, Y., Mao, S., Mei, J., Men, X., Miao, Y., Pan, S., Peng, Y., Qin, R., Qin, Z., Qu, B., Shang, Z., Shi, L., Shi, S., Song, F., Su, J., Su, Z., Sui, L., Sun, X., Sung, F., Tai, Y., Tang, H., Tao, J., Teng, Q., Tian, C., Wang, C., Wang, D., Wang, F., Wang, H., Wang, H., Wang, J., Wang, J., Wang, J., Wang, S., Wang, S., Wang, S., Wang, X., Wang, Y., Wang, Y., Wang, Y., Wang, Y., Wang, Y., Wang, Z., Wang, Z., Wang, Z., Wei, C., Wei, Q., Wu, H., Wu, W., Wu, X., Wu, Y., Xiao, C., Xie, J., Xie, X., Xiong, W., Xu, B., Xu, J., Xu, L., Xu, S., Xu, W., Xu, X., Xu, Y., Xu, Z., Xu, J., Yan, J., Yan, Y., Yang, H., Yang, X., Yang, Y., Yang, Y., Yang, Z., Yang, Z., Yang, Z., Yao, H., Yao, X., Ye, W., Ye, Z., Yin, B., Yu, L., Yuan, E., Yuan, H., Yuan, M., Yuan, S., Zhan, H., Zhang, D., Zhang, H., Zhang, W., Zhang, X., Zhang, Y., Zhang, Y., Zhang, Y., Zhang, Y., Zhang, Y., Zhang, Y., Zhang, Y.,

Zhang, Y., Zhang, Z., Zhao, H., Zhao, Y., Zhao, Z., Zheng, H., Zheng, S., Zhong, L., Zhou, J., Zhou, X., Zhou, Z., Zhu, J., Zhu, Z., Zhuang, W., Zu, X.: Kimi K2: Open Agentic Intelligence (2026). https://arxiv.org/abs/2507.20534

[22] MiniMax, Chen, A., Li, A., Gong, B., Jiang, B., Fei, B., Yang, B., Shan, B., Yu, C., Wang, C., Zhu, C., Xiao, C., Du, C., Zhang, C., Qiao, C., Zhang, C., Du, C., Guo, C., Chen, D., Ding, D., Sun, D., Li, D., Jiao, E., Zhou, H., Zhang, H., Ding, H., Sun, H., Feng, H., Cai, H., Zhu, H., Sun, J., Zhuang, J., Cai, J., Song, J., Zhu, J., Li, J., Tian, J., Liu, J., Xu, J., Yan, J., Liu, J., He, J., Feng, K., Yang, K., Xiao, K., Han, L., Wang, L., Yu, L., Feng, L., Li, L., Zheng, L., Du, L., Yang, L., Zeng, L., Yu, M., Tao, M., Chi, M., Zhang, M., Lin, M., Hu, N., Di, N., Gao, P., Li, P., Zhao, P., Ren, Q., Xu, Q., Li, Q., Wang, Q., Tian, R., Leng, R., Chen, S., Chen, S., Shi, S., Weng, S., Guan, S., Yu, S., Li, S., Zhu, S., Li, T., Cai, T., Liang, T., Cheng, W., Kong, W., Li, W., Chen, X., Song, X., Luo, X., Su, X., Li, X., Han, X., Hou, X., Lu, X., Zou, X., Shen, X., Gong, Y., Ma, Y., Wang, Y., Shi, Y., Zhong, Y., Duan, Y., Fu, Y., Hu, Y., Gao, Y., Fan, Y., Yang, Y., Li, Y., Hu, Y., Huang, Y., Li, Y., Xu, Y., Mao, Y., Shi, Y., Wenren, Y., Li, Z., Li, Z., Tian, Z., Zhu, Z., Fan, Z., Wu, Z., Xu, Z., Yu, Z., Lyu, Z., Jiang, Z., Gao, Z., Wu, Z., Song, Z., Sun, Z.: MiniMax-M1: Scaling Test-Time Compute Efficiently with Lightning Attention (2025). https://arxiv.org/abs/2506.13585

[23] Team, G., Zeng, A., Lv, X., Zheng, Q., Hou, Z., Chen, B., Xie, C., Wang, C., Yin, D., Zeng, H., Zhang, J., Wang, K., Zhong, L., Liu, M., Lu, R., Cao, S., Zhang, X., Huang, X., Wei, Y., Cheng, Y., An, Y., Niu, Y., Wen, Y., Bai, Y., Du, Z., Wang, Z., Zhu, Z.: GLM-4.5: Agentic, Reasoning, and Coding (ARC) Foundation Models (2025). https://arxiv.org/abs/2508.06471

[24] Karpukhin, V., Oguz, B., Min, S., Lewis, P., Wu, L., Edunov, S., Chen, D., Yih, W.-t.: Dense Passage Retrieval for Open-Domain Question Answering (2020). https://arxiv.org/abs/2004.04906

[25] Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W.-t., Rocktäschel, T., Riedel, S., Kiela, D.: Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks (2020). https://arxiv.org/abs/2005.11401

[26] Zhang, Y., Li, Y., Cui, L., Cai, D., Liu, L., Fu, T., Huang, X., Zhao, E., Zhang, Y., Xu, C., Chen, Y., Wang, L., Luu, A.T., Bi, W., Shi, F., Shi, S.: Siren's Song in the AI Ocean: A Survey on Hallucination in Large Language Models (2025). https://arxiv.org/abs/2309.01219

[27] Farquhar, S., Kossen, J., Kuhn, L., Gal, Y.: Detecting hallucinations in large language models using semantic entropy. Nature **630**(8017), 625–630 (2024) https://doi.org/10.1038/s41586-024-07421-0

[28] Nakano, R., Hilton, J., Balaji, S., Wu, J., Ouyang, L., Kim, C., Hesse, C., Jain, S., Kosaraju, V., Saunders, W., Jiang, X., Cobbe, K., Eloundou, T., Krueger,

G., Button, K., Knight, M., Chess, B., Schulman, J.: WebGPT: Browser-assisted Question-Answering with Human Feedback (2021). https://arxiv.org/abs/2112.09332

[29] Schick, T., Dwivedi-Yu, J., Dessì, R., Raileanu, R., Lomeli, M., Hambro, E., Zettlemoyer, L., Cancedda, N., Scialom, T.: Toolformer: Language Models Can Teach Themselves to Use Tools (2023). https://arxiv.org/abs/2302.04761

[30] Qin, Y., Liang, S., Ye, Y., Zhu, K., Yan, L., Lu, Y., Lin, Y., Cong, X., Tang, X., Qian, B., Zhao, S., Tian, R., Xie, R., Zhou, J., Gerstein, M., Li, D., Liu, Z., Sun, M.: ToolLLM: Facilitating Large Language Models to Master 16000+ Real-world APIs (2023). https://arxiv.org/abs/2307.16789

[31] Yao, S., Zhao, J., Yu, D., Du, N., Shafran, I., Narasimhan, K., Cao, Y.: ReAct: Synergizing Reasoning and Acting in Language Models (2023). https://arxiv.org/abs/2210.03629

[32] Du, Y., Ye, R., Tang, S., Zhu, X., Lu, Y., Cai, Y., Chen, S.: OpenSeeker: Democratizing Frontier Search Agents by Fully Open-Sourcing Training Data (2026). https://arxiv.org/abs/2603.15594

[33] Du, Y., Ye, R., Tang, S., Huang, K., Zhu, X., Cai, Y., Chen, S.: Openseeker-v2: Pushing the limits of search agents with informative and high-difficulty trajectories. arXiv preprint arXiv:2605.04036 (2026)

[34] Guo, T., Chen, X., Wang, Y., Chang, R., Pei, S., Chawla, N.V., Wiest, O., Zhang, X.: Large Language Model Based Multi-Agents: A Survey of Progress and Challenges (2024). https://arxiv.org/abs/2402.01680

[35] Shinn, N., Cassano, F., Gopinath, A., Narasimhan, K., Yao, S.: Reflexion: Language Agents with Verbal Reinforcement Learning (2023). https://arxiv.org/abs/2303.11366

[36] Kang, H., Xiong, C.: ResearchArena: Benchmarking Large Language Models' Ability to Collect and Organize Information as Research Agents (2025). https://arxiv.org/abs/2406.10291

[37] Bragg, J., D'Arcy, M., Balepur, N., Bareket, D., Dalvi, B., Feldman, S., Haddad, D., Hwang, J.D., Jansen, P., Kishore, V., Majumder, B.P., Naik, A., Rahamimov, S., Richardson, K., Singh, A., Surana, H., Tiktinsky, A., Vasu, R., Wiener, G., Anastasiades, C., Candra, S., Dunkelberger, J., Emery, D., Evans, R., Hamada, M., Huff, R., Kinney, R., Latzke, M., Lochner, J., Lozano-Aguilera, R., Nguyen, C., Rao, S., Tanaka, A., Vlahos, B., Clark, P., Downey, D., Goldberg, Y., Sabharwal, A., Weld, D.S.: AstaBench: Rigorous Benchmarking of AI Agents with a Scientific Research Suite (2026). https://arxiv.org/abs/2510.21652

[38] Xiong, L., Luo, K., Xia, Z., Zhang, W., Yao, J.-G., Liu, Z., Shao, J., Chen, J.,

Qian, H., Yang, X., Yu, Q., Li, H., Yue, C., Du, X., Wang, Y., Liu, Y., Xu, H., Dou, Z.: AutoResearchBench: Benchmarking AI Agents on Complex Scientific Literature Discovery (2026). https://arxiv.org/abs/2604.25256

[39] Manning, C.D., Raghavan, P., Schütze, H.: Introduction to Information Retrieval. Cambridge University Press, Cambridge (2008). https://doi.org/10.1017/CBO9780511809071

[40] Järvelin, K., Kekäläinen, J.: Cumulated gain-based evaluation of ir techniques. ACM Trans. Inf. Syst. **20**(4), 422–446 (2002) https://doi.org/10.1145/582415.582418

[41] Lo, K., Wang, L.L., Neumann, M., Kinney, R., Weld, D.S.: S2ORC: The Semantic Scholar Open Research Corpus (2020). https://arxiv.org/abs/1911.02782

[42] Thakur, N., Reimers, N., Rücklé, A., Srivastava, A., Gurevych, I.: BEIR: A Heterogeneous Benchmark for Zero-shot Evaluation of Information Retrieval Models (2021). https://arxiv.org/abs/2104.08663